(12) UK Patent Application (19) GB (11) 2 314 948 (13) A

(43) Date of A Publication 14.01.1998

(21) Application No 9712374.9

(22) Date of Filing 14.06.1997

(30) Priority Data
(31) 19626339 (32) 01.07.1996 (33) DE

(71) Applicant(s)
International Business Machines Corporation

(Incorporated in USA - New York)

Armonk, New York 10504, United States of America

(72) Inventor(s)
Walter Hänel

(74) Agent and/or Address for Service
P Waldner
IBM United Kingdom Limited, Intellectual Property Department, Hursley Park, WINCHESTER, Hampshire, SO21 2JN, United Kingdom

(51) INT CL^6
G06F 12/14 9/445

(52) UK CL (Edition P)
G4A AAP AFL

(56) Documents Cited
WO 87/07061 A1 US 4853522 A

(58) Field of Search
UK CL (Edition O) G4A AAP AMG1
INT CL^6 G06F 1/00 12/02 12/06 12/14 15/02
ONLINE: WPI, INSPEC, COMPUTER

(54) Secure transfer of applications and/or data onto a chipcard

(57) Secure loading of different applications onto a chipcard (IC card) which has already been issued. The chipcard has a memory with a file structure comprising a plurality of subdomains. A terminal sends at least one command to the chipcard which can be a CREATE_FILE command, shown, to create new files and new file directories in the existing file structure, or a command to read and/or write data to the memory. The use of the command is limited to particular subdomains by means of an identifier allocated to the command, where the identifier comprises identification data containing information on the file directories in which the command can be used. Alternatively, if several commands are to be certified for a particular subdomain, an identifier can be allocated to the subdomain (fig. 3). Security codes such as cryptographic keys or passwords can similarly be limited to particular subdomains.

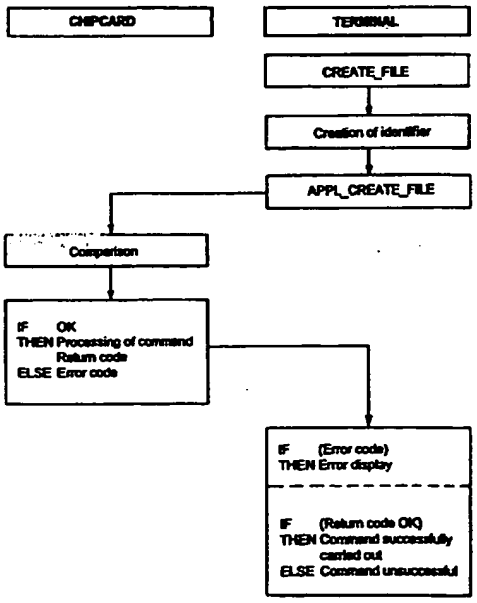

FIG. 2

GB 2 314 948 A

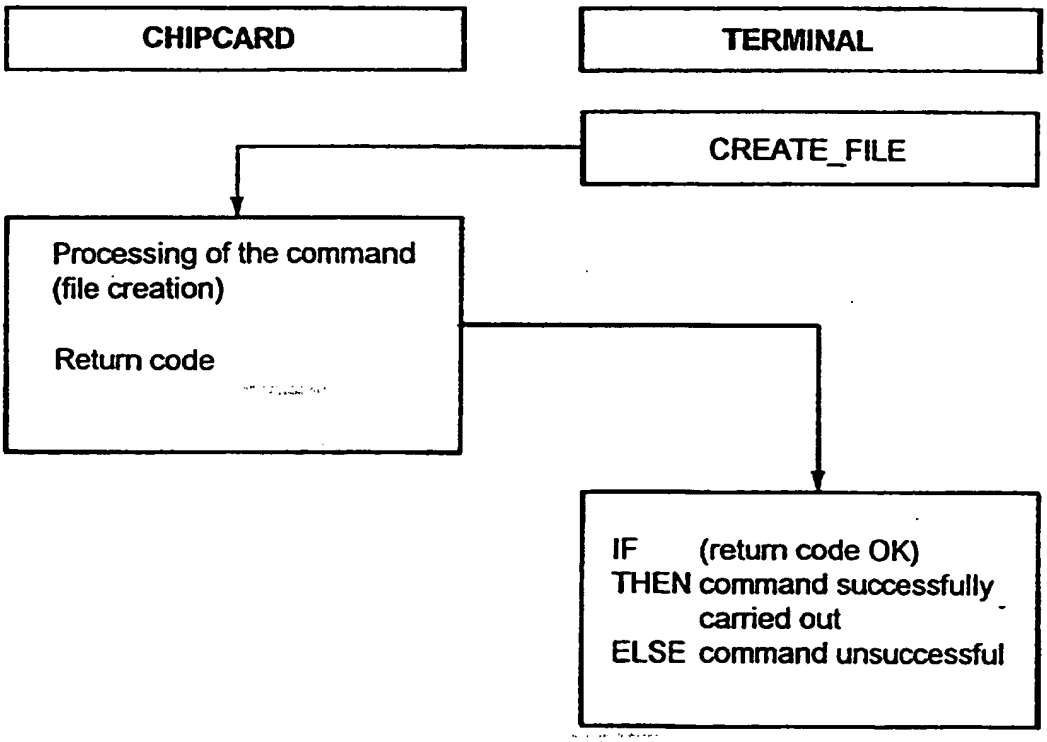
CHIPCARD
TERMINAL
CREATE_FILE
Processing of the command
(file creation)
Return code
IF (return code OK)
THEN command successfully
carried out
ELSE command unsuccessful

FIG. 1

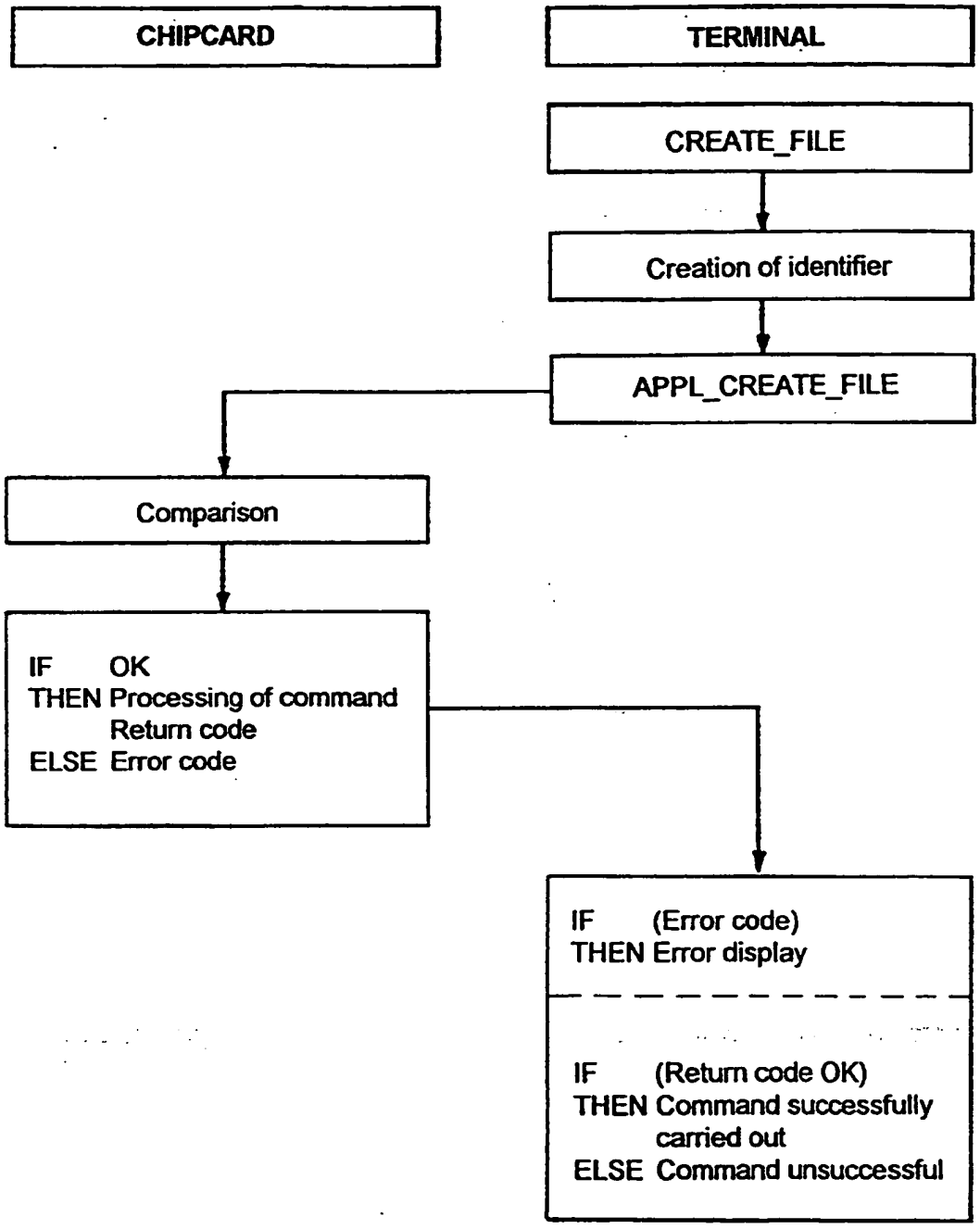
CHIPCARD
TERMINAL
CREATE_FILE
Creation of identifier
APPL_CREATE_FILE
Comparison
IF OK
THEN Processing of command
Return code
ELSE Error code
IF (Error code)
THEN Error display
IF (Return code OK)
THEN Command successfully
carried out
ELSE Command unsuccessful

FIG. 2

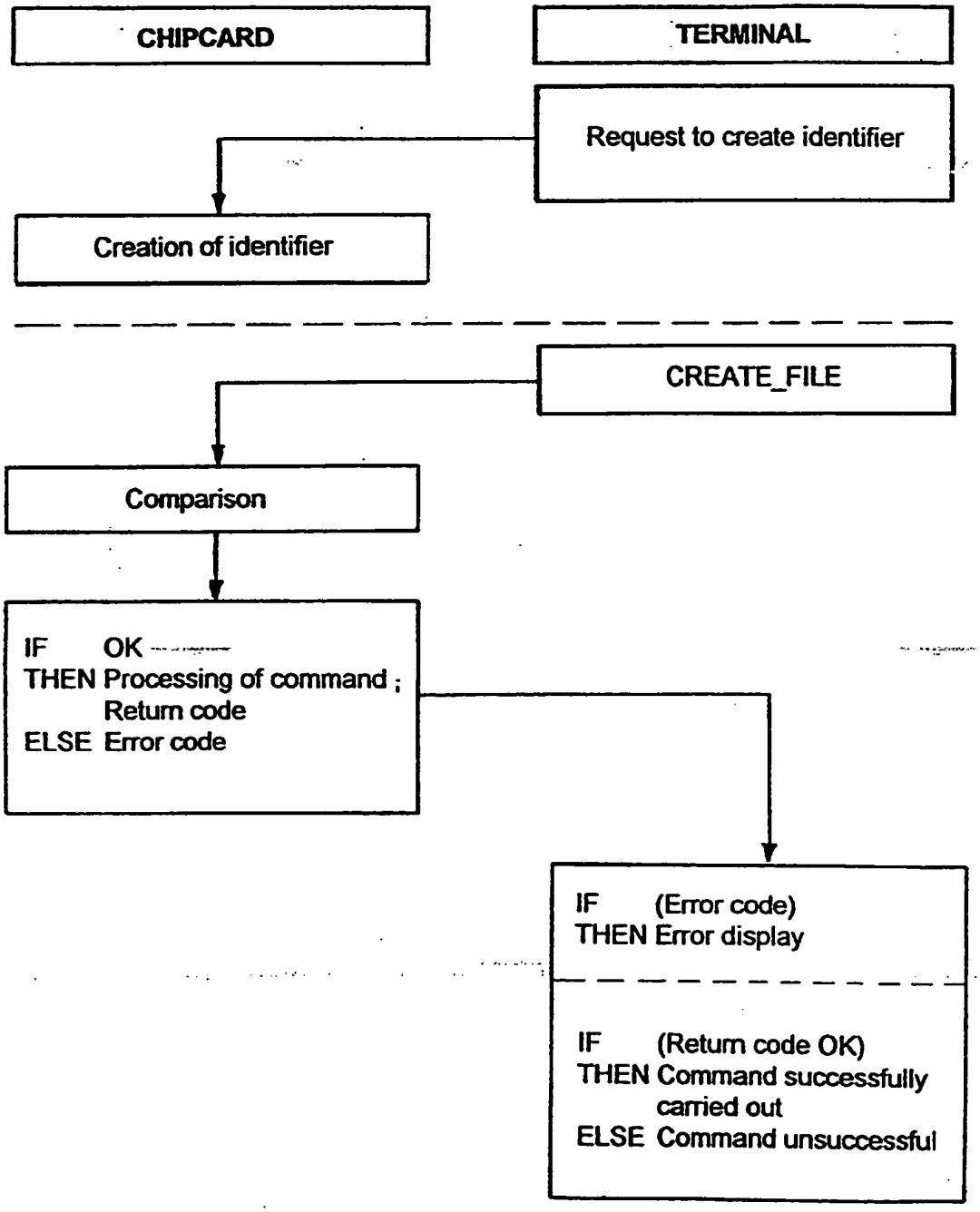
CHIPCARD
TERMINAL
Request to create identifier
Creation of identifier
CREATE_FILE
Comparison
IF OK
THEN Processing of command ;
Return code
ELSE Error code
IF (Error code)
THEN Error display
IF (Return code OK)
THEN Command successfully
carried out
ELSE Command unsuccessful

FIG. 3

CHIPCARD DATA TRANSFER METHOD

The invention offers the secure transfer of applications and/or data onto a chipcard which has a memory with a file structure and where, using a process by means of at least one command, a changed file structure is created in the memory and/or data are read from the memory and/or data are written to the memory.

Since the mid eighties, chipcard applications have increased in areas of everyday life. This success is based fundamentally on their high levels of manipulation security code and their reliability. In addition, chip technology guarantees high levels of flexibility for a number of applications.

The manufacture of chipcards up to the point where they can be issued to a user is described in Rankl/Effing: Handbuch der Chipkarten (The Chipcard Handbook), Karl Hanser Verlag, 1996. After a module with the semi-conductor chip is embedded into the card, global data and personal data on the future card user is then loaded onto the chipcard. Through this, manufacturers of chipcards are increasingly frequently loading several applications onto the chipcard at the same time.

The internal structure of the chipcard basically conforms to the ISO 7816-4 norm. Usually, the data which belong to an application are placed in files which can be found in an application directory. The files and application directories are transferred by the card manufacturer onto the chip card. If a new application from an application supplier is now to be transferred onto a chipcard already issued, then particular attention must be paid to the chipcard security code system. This particularly applies to applications which are not under the control of the card manufacturer.

If one supposes that the application supplier has an interest in spying on data from the manufacturer or other application suppliers or using their security code systems, in particular cryptographic keys, then the task involves in preventing this spying by the application supplier.

In a known procedure, commands which could be used to create files and application directories on the chipcard are certified by the manufacturer of the chipcard. This is done by, for example, appending a MAC - message authentication code. In this, using a cryptographic key

belonging to the manufacturer, an individual MAC is added to the command to be used and appended to the command. This process is carried out by a cryptographic installation which the manufacturer has made available to the application supplier. The installation converts the command into a form acceptable to the chipcard.

The disadvantage of this known procedure is that the command created by the cryptographic installation can still be used in a directory in which its use had not been foreseen. This is possible as the commands rely on previous selection commands with unlimited access to different directories. Thus data and/or cryptographic keys belonging to the manufacturer or other application suppliers can be overwritten or changed.

There is an additional problem when applications from one application supplier are to be transferred at a later date onto a chipcard already issued. If the manufacturer allows the application supplier to create new files, this can be exploited by the application supplier in order to spy on cryptographic keys. When creating new files and the rights of access associated with them, it is possible to place the rights of access in such a way that on reading data from the file and/or writing data to a file, reference is made to the manufacturer's cryptographic key. In this, the manufacturer of the chipcard can be misrepresented as being the sender of data, although it was actually the application supplier.

Therefore, the task of the invention presented is to create the possibility of securely loading applications onto a chipcard already issued.

This task is solved in one embodiment of the present invention by the command only being used in at least one subdomain of a common file structure, where the file structure and the changed file structure are included in the common file structure.

According to one aspect of the present invention there is provided method for the secure transfer of applications and/or data onto a chipcard having a memory with a common file structure, having a plurality of subdomain the method comprising the steps of:

creating a changed file structure in the memory and/or reading data from the memory and/or writing data to the memory by means of at least one command; wherein

the command is used only in at least one subdomain of the common file structure; and

the file structure and the changed file structure is covered by the common file structure.

According to a second aspect of the present invention there is provided a chipcard having a memory with a basic file structure, said chipcard comprising:

of at least one command on the clipboard for loading an expandable file structure in the memory;

data means for reading data from the memory;

means for writing data to the memory; and

security code means, in particular a cryptographic key and/or a password;

the use of the command and/or the security code to at least one subdomain in the common file structure;

where the common file structure consists of the basic structure and the expandable file structure.

The most important advantage achieved by at least one embodiment of the invention with regard to the current state of technology is the guarantee of secure loading of a multitude of different applications onto a chipcard which has already been issued and where the applications of independent application suppliers can be loaded. Each application supplier can only use the command certified for him by the manufacturer in a certain area of the chipcard and thus does not have the possibility of accessing subdomains of the chipcard memory which are reserved either for other application suppliers or exclusively for the manufacturer.

A further advantage of at least one embodiment of the invention is that by limiting the usability of cryptographic keys, their use can be controlled by the chipcard manufacturer.

In one embodiment of the invention at least one level of security code is used, in particular a cryptographic key and/or a password. The use of this type of security code guarantees improved security code standards on transferring applications onto the chipcard.

Advantages can be seen in that the security code is only used in at least one subdomain of the file structure, where the risk of erroneous use of security code in other subdomains of the common file structure is reduced.

On proper continuation of at least one embodiment of the invention, the command and/or security code will each have at least one identifier allocated to it. An identifier has the advantage that it can be flexibly adapted to the respective conditions of use. If the manufacturer wants to make available a certain command and/or security code to several application suppliers, then this can be done using the identifier without great effort.

Advantages can be seen in that the identifier shows whether the command and/or security code can be used in the subdomain of the common file structure. This helps avoid the misuse of commands and/or security code in application directories of the chipcard where their use has not been foreseen.

One advantageous form of at least one embodiment of the invention appears in the identifier of the command and/or the identifier of the security code being given before the command and/or the security code is used in the subdomain of the common file structure. This stage in the process opens up the option of deciding on subsequent procedural stages which are dependent on the identifier, before the command and/or the security code being used is started.

The use of the command and/or security code in the subdomain of the common file structure can be purposely excluded when the identifier of the command and/or security code shows that the command and/or security code cannot be used in the subdomain of the common file structure. In this, the misuse of a command and/or a security code in the subdomain can be avoided. This is carried out advantageously before any data manipulation can be started by means of the command and/or security code.

In the dependent claims 8 to 11, the advantages stated in connection with the dependent claims 4 to 7 apply. Particularly advantageous is the allocation of the identifier to the subdomain of the common file structure, if the manufacturer wants to certify for an application supplier several commands and/or security codes for this subdomain. In this case, one identifier with the necessary information

will be allocated once to the subdomain. This saves a number of procedural stages in the identification of the several commands and/or security codes.

In an advantageous continuation of at least one embodiment of the invention, the command and/or security code is located in the subdomain of the common file structure where the command and/or the security code can be stored in the subdomain in which their possible use has been foreseen.

In the dependent claims 13 to 15, the advantages stated in connection with the dependent claims 5 to 7 apply.

In the dependent claims 16 to 18, the advantages stated in connection with the dependent claims 9 to 11 apply.

In the claims 19 to 21, the advantages stated in connection with the associated claims apply.

In order to promote a fuller understanding of this and other aspects of the invention, an embodiment will now be described, by way of example only, with reference to the accompanying drawings in which:

Fig. 1a schematic representation of a known process to create a file in an existing file structure on a chipcard

Fig. 2 a schematic representation of the invention process to create a file in an existing file structure on a chipcard, where the command APPL_CREATE_FILE has the identification data and

Fig. 3 a schematic representation of the invention process to create a file in an existing file structure on a chipcard, where the directory APPL_2 has the identification.

The communication procedure between a terminal and a chipcard is fundamentally based on the so-called "request for response procedure". This means that the terminal sends a command to the chipcard as a "request", and the latter then processes it and creates a response and returns this to the terminal as the "response". Therefore the card does not send any data without having received the command to do so from the terminal. The terminal is mainly a writing and/or reading device for chipcards.

One of the most important commands which is used in the transferring of applications in the chipcard memory at a later date is the CREATE command. Using this, new files and new file directories can be created, in particular in an existing file structure on the chipcard. By using the CREATE command to create a new file (CREATE_FILE), basically the following data are transferred: the type of the file to be created, application identification, file identification, access conditions and structure of the new file, size of the file, number of records, and length of the records or length of each individual record.

In figure 1, the course of a known process for creating a new file in an existing file structure on a chipcard is represented schematically. The command CREATE_FILE is sent to the chipcard with parameters and data. The new file is created on the chipcard, where the characteristics (size, type, ...) of the new file are dependent on the command parameters. The chipcard then sends back a return code to the terminal if the CREATE_FILE command has been correctly carried out.

In a preferred design of the invention process, an identifier is first allocated to the CREATE_FILE command. The command should then be called APPL_CREATE_FILE. The issuing of an identifier can be carried out by means of appending identification data to the existing data set of the CREATE_FILE command.

The identification data contain information on the file directory in which the APPL_CREATE_FILE command can be used. The file directory covers any subdomain of at least one file in the existing common file structure on the chipcard which has a number of files. The identification data can be carried out mainly using a data bit. The data bit has the value of "1" or "0" according to whether the APPL_CREATE_FILE command may be used in the directory or not.

The identification data can also contain the name of the file directory in which the APPL_CREATE_FILE command may be applied.

A data bit sequence can also be loaded into the identification data. This is particularly advantageous if one wishes to allow the use of a certain command in several data directories.

Fig. 2 shows a schematic representation of the invention procedure. Firstly, the APPL_CREATE_FILE command including the identification data

is defined in the terminal. This command is then sent to the chipcard in order to create a file in file directory APPL_1. A comparison is now carried out on the chipcard. Using the identification data of the received APPL_CREATE_FILE command, it is determined whether the received APPL_CREATE FILE command may be used in the APPL_1 file directory. If the result of the comparison is positive, i.e. the received APPL_CREATE_FILE command may be used in the APPL_1 file directory, then a new file is created and then a return code is sent back to the terminal. In the case of a negative result, further processing of the command received will be stopped and an error code is sent back to the terminal. Based on this error code, an error message can be created in an output device at the terminal, in order to inform the terminal user of the unsuccessful processing of the APPL_CREATE_FILE command.

The invention is not limited to commands to change the file structure of the chipcard. The issuing of a command identifier and stages following on from this can also be carried out in connection with commands to read and/or write data to the chipcard.

As an alternative to the invention design described, an identifier can also be linked with a file directory, i.e. with a subdomain of the file structure on the chipcard. In this case, the file directory has the identifier, and not the command to be carried out. The identifier of the file directory contains information on which commands may be used in this file directory.

The invention procedure in the case of a file directory identifier is described in the following using the example of the CREATE_FILE command. As will be seen, a definition of the new command APPL_CREATE_FILE is not needed. After the command CREATE_FILE is sent to the chipcard (fig. 3), in order to create a new file in the file directory APPL_2, the identification of the directory APPL_2 is issued to the chipcard and it is determined whether the command CREATE_FILE may be used in the directory APPL_2. The command received is processed if the identifier so allows. If this is not the case, then the processing of the command CREATE_FILE is not started and an error code is sent back to the terminal in order to finally inform the user of the unsuccessful execution of the command in file directory APPL_2.

Very often, security codes which are mainly cryptographic keys or passwords are used in order to satisfy the security code requirements in

loading data onto a chipcard. Here, using the invention procedure, use limitation checks are possible, i.e. a limit to the use of a certain security code to one or several file directories. The use limitation check, like the previously described command use check, is achieved either by adding an identifier to the security code or by means of an identifier in the file directory in which the security code is to be used.

The course of the invention procedure, after receiving a request from the terminal to use a security code on the chipcard, conforms, in this case, to the course described in the previous sections, in connection with the command use control. Using the identifier, before a security code is used in a special file directory, the security code and/or identifier of the file directory is checked to see whether its use in this special file directory is permitted and, finally, either it is possible to use the security code, or an error message is sent to the terminal.

In another preferred design of the invention, the commands CREATE_FILE and/or APPL_CREATE_FILE are allocated to the APPL_3 file directory. If the chipcard now receives a request from the terminal to use this command to create a file in a directory outside the APPL_3 directory, then the invention procedure makes sure that the commands CREATE_FILE and/or APPL_CREATE_FILE are only used in directories outside APPL_3, in which they are allowed. In order to achieve this in the case of the command APPL_CREATE_FILE, this command is equipped with identification data. It can be inferred from these identification data whether the command APPL_CREATE_FILE may be used in a directory other than APPL_3, and if so, in which.

In the alternative design, whether the command CREATE_FILE may be used outside APPL_3 is based on the identifier of the APPL_3 directory. If this is not allowed, according to the identifier, then a corresponding request received by the terminal will not be carried out. An error message will be sent back to the terminal.

The embodiments described in the last two sections, characterised by the command to be used being allocated in a file directory on the chipcard, can be transferred without great effort to security codes. The security codes can also be arranged in a directory. The check on the use of these security codes outside the directories in which they have been

filed is carried out in connection with the command according to the described procedures.

In summary, there is described the secure loading of applications and/or data onto a chipcard having a memory with a file structure. By using at least one command a changed file structure is created in the memory data is read from the memory data is written to the memory. As well as commands, security codes can also be used. The use of the commands and/or security codes is limited by means of the identifiers being issued to subdomains in the file structure of the chipcard.

In a further development of the invention, the use control of certain commands and/or security codes can be carried out by means of a combination of identifiers on a command and/or security code and identifiers in a directory.

CLAIMS

1. Method for the secure transfer of applications and/or data onto a chipcard having a memory with a common file structure, having a plurality of subdomain the method comprising the steps of:

creating a changed file structure in the memory and/or reading data from the memory and/or writing data to the memory by means of at least one command; wherein

the command is used only in at least one subdomain of the common file structure; and

the file structure and the changed file structure is covered by the common file structure.

2. The method according to claim 1, whereby;

at least one security code is used, in particular a cryptographic key and/or password.

3. The method according to claim 2, whereby;

the security code only is used in at least one subdomain of the file structure.

4. The method according to claims 1 or 3, whereby;

the command and/or security code each have at least one identifier allocated to them.

5. The method according to claim 4, whereby;

it is shown by means of the identifier whether the command and/or security code can be used in the subdomain of the common file structure.

6. The method according to claim 5, whereby;

the identifier of the command and/or the identifier of the security code is established before the command and/or the security code is used in the subdomain of the common file structure.

7. The method according to claim 6, whereby;

the use of the command and/or security code in the subdomain of the common file structure is excluded in the case of the identifier of the command and/or the security code shows that the command and/or the security code may not be used in the subdomain of the common file structure.

8. The method according to claims 1 or 3, whereby;

the subdomain of the common file structure has at least one identifier allocated to it.

9. The method according to claim 8, wherein;

it is shown by means of the identifier of the subdomain of the common file structure whether the command and/or security code can be used in the subdomain of the common file structure.

10. The method according to claim 9, wherein;

the identifier of the subdomain of the common file structure is established before the command and/or the security code is used in the subdomain of the common file structure.

11. The method according to claim 10, whereby;

the use of the command and/or security code in the subdomain of the common file structure is excluded in the case of the identifier of the subdomain showing that the command and/or the security code may not be used in the subdomain of the common file structure.

12. The method according to claims 1 or 3, whereby;

the command and/or the security code is allocated in the subdomain of the common file structure.

13. The method according to claims 4 and 12, whereby;

it is shown by means of the identifier of the command and/or security code whether the command and/or security code can be used outside the subdomain of the common file structure.

14. The method according to claim 13, whereby;

the identifier of the command and/or the identifier of the security code is established before the command and/or the security code is used outside the subdomain of the common file structure.

15. The method according to claim 14, whereby;

the use of the command and/or security code outside the subdomain of the common file structure is excluded in the case of the identifier of the command and/or the security code shows that the command and/or the security code may not be used outside the subdomain of the common file structure.

16. The method according to claims 8 and 12, whereby;

it is shown by means of the identifier of the subdomain whether the command and/or security code can be used outside the subdomain of the common file structure.

17. The method according to claim 16, whereby;

the identifier of the subdomain is established before the command and/or the security code is used outside the subdomain of the common file structure.

18. The method according to claim 17, whereby;

the use of the command and/or security code outside the subdomain of the common file structure is excluded in the case of the identifier of the subdomain showing that the command and/or the security code may not be used outside the subdomain of the common file structure.

19. A chipcard having a memory with a basic file structure, said chipcard comprising:

of at least one command on the clipboard for loading an expandable file structure in the memory;

data means for reading data from the memory;

means for writing data to the memory; and

security code means, in particular a cryptographic key and/or a password;

the use of the command and/or the security code to at least one subdomain in the common file structure;

where the common file structure consists of the basic structure and the expandable file structure.

20. A chipcard according to claim 19, wherein;

the security code means is designed as at least one identifier for the command and/or at least one identifier for the security code.

21. A chipcard according to claim 19, wherein;

the security code means is designed as at least one identifier for the subdomain of the common file structure.

22. A method as substantially described herein with reference to figures 2 and 3.

23. A chip card for storing application files comprising:

memory having file memory for file data storage and directory memory for file reference storage, said directory memory comprising a plurality of sub-domain memories for grouping particular file references;

means for receiving an instruction to store a file in said memory, said instruction including a sub-directory reference for said file;

means for checking whether the instruction is a valid instruction for use in the referenced sub-directory;

means for storing, on receipt of a valid instruction, said file in a location in memory;

means for storing, on receipt of a valid instruction, said file reference in the referenced sub-directory.

The Patent Office

15

Application No:	GB 9712374.9	**Examiner:**	Melanie Jennings
Claims searched:	1-18	**Date of search:**	18 August 1997

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.O): G4A (AAP, AMG1)

Int Cl (Ed.6): G06F 1/00, 12/02, 12/06, 12/14, 15/02

Other: Online: WPI, INSPEC, COMPUTER

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	WO 87/07061 A1	(AMERICAN TELEPHONE & TELEGRAPH), see especially pages 8 - 10.	1 - 3
X	US 4853522 A	(OGASAWARA), see whole document.	1 - 3

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.